

# 第十五节 Spring 运维架构重构

---

## Spring 配置

---

### Profiles

Spring 应用启动过程中激活的配置文件内容，通过逻辑比较判断，那部分组件是否能够在其指定 Profiles 下激活。

### PropertySources

#### 核心 API

##### PropertySource 和 @PropertySource

PropertySource 类似于 MicroProfile Config 中的 ConfigSource

@PropertySource

**PropertySources 和 @PropertySources**

# Spring JMX

---

## 核心 API

**JMX Bean 暴露器 -**

**org.springframework.jmx.export.MBeanExporter**

实现接口

- Aware 接口
  - BeanClassLoaderAware - 注入记载 Bean 的 ClassLoader
  - BeanFactoryAware - 注入 BeanFactory
- Bean 对象生命周期接口
  - InitializingBean
  - SmartInitializingSingleton
  - DisposableBean

# Spring Beans JSON Viewer - LiveBeansView

```
[
  {
    "context":
      "org.springframework.context.annotation.AnnotationConfigApplicationContext@4b85612c",
    "parent": null,
    "beans": [
      {
        "bean": "springJmxDemo",
        "aliases": [],
        "scope": "singleton",
        "type":
          "org.geektimes.spring.jmx.SpringJmxDemo$$EnhancerBySpringCGLIB$$d87790a9",
        "resource": "null",
        "dependencies": []
      },
      {
        "bean": "beanExporter",
        "aliases": [],
        "scope": "singleton",
        "type":
          "org.springframework.jmx.export.annotation.AnnotationMBeanExporter",
        "resource":
          "org.geektimes.spring.jmx.SpringJmxDemo",

```

```
        "dependencies": [
            "springJmxDemo"
        ]
    },
    {
        "bean": "user",
        "aliases": [],
        "scope": "singleton",
        "type":
"org.geektimes.spring.jmx.User",
        "resource":
"org.geektimes.spring.jmx.SpringJmxDemo",
        "dependencies": [
            "springJmxDemo"
        ]
    },
    {
        "bean": "liveBeansView",
        "aliases": [],
        "scope": "singleton",
        "type":
"org.springframework.context.support.LiveBeansV
iew",
        "resource":
"org.geektimes.spring.jmx.SpringJmxDemo",
        "dependencies": [
            "springJmxDemo"
        ]
    }
]
}
```

# Spring Logging

---

Spring repackages

- Apache Commons Logging
- CGLIB
- ASM
- AOP Alliance
- ...

减少版本冲突

- spring-context
  - spring-beans
    - spring-core

## Spring 熟悉的 jar

spring-core

spring-beans

spring-context

# 相关技术

---

## 通用 Java 对象生命周期

### 1. 实例化阶段

- 编码方式: `new XXX(...)`
- 反射方式: `XXX.class.newInstance(...)`

### 2. 属性被填充 (赋值) 阶段

### 3. 初始化阶段

- `@PostConstruct`
- `Spring InitializingBean`
- 自定义 `init` 方法

## MicroProfile Config

### 配置源 - ConfigSource

## JMX

Jolokia - REST bridges JMX

## 四种 MBean

- 标准 MBean
  - MBean 类名结尾的接口
  - MXBean (@MXBean)
- 动态 MBean - 没有固定接口结构
- 开放 MBean - 没有特定类型
- 模型 MBean - 更大范围对象类型适配 (模型 POJO)

结论：所有的 JMX 类型 MBean 都是 DynamicMBean 实现 (底层)

## 本周作业

---

- 使用 Spring Boot 来实现一个整合Gitee/或者Github OAuth2 认证
  - Servlet